CLAIMS:

What is claimed is:

1. A method of optimizing memory resources, comprising:

establishing a history file recording a data processing criterion;

storing said history file in a first memory;

selectively reallocating information residing in or to be stored in said first memory to an alternate memory;

retrieving said reallocated information in a portion based on said data processing criterion established in said history file; and

restoring said reallocated information back to said first memory.

2. The method of claim 1, further comprising the step of updating said history file after selective time periods to include most recent data processing information.

3. The method of claim 2, wherein said criterion includes rate of information retrieval.

4. The method of claim 3, wherein said history file includes historical processing rates for information retrieval.

5. The method of claim 4, wherein said historical processing rate is calculated by measuring historical a peak rate during which information processing rate is at its highest rate over a certain time period.

6. The method of claim 5, wherein said historical peak processing rate also records occasional spikes in capacity.

7. The method of claim 6, wherein said peak processing rate is updated frequently so that an old occasional spike record is not stored indefinitely.

8. The method of claim 5, wherein each time an application accesses a file residing in said memory during a sampling period, a count is established and incremented to track number of accesses made to said file.

9. The method of claim 8, further comprising the steps of ending sampling period when said count reaches a certain value; and calculating a sample access rate based on rate and frequency that said application accesses said file; and storing said access sample rate in said history file to be used later for information retrieval.

10. The method of claim 9, wherein said peak rates compared to said sample access rate and the higher of said two values is stored in said history file as a new peak rate for later retrieval.

11. The method of claim 9, wherein if said peak rate will be adjusted if it is obtained prior to a certain time period.

12. The method of claim 10, wherein said restoration of said information is performed by retrieving said new peak rate from said history file and comparing it to a most current processing rate, the higher of the two value establishing an information retrieval rate.

13. The method of claim 12, wherein said information retrieval rate is highest of three values of said new peak rate, said most current processing rate an a third user override rate.

14. The method of claim 1, wherein said retrieval rate is used to determine how much of said data can be restored to said memory during each processing period.

15. The method of claim 2, wherein if only a portion of said reallocated data can be restored during a processing period, remainder of said reallocated information will be restored to said memory during one or more new processing periods by using said history file each time to determine how much information can be transferred during each new processing period so that said memory is

never under-utilized or overwhelmed by too little information or too much information transfer during any single processing period.

16. The method of claim 1, where a plurality of processing criteria are established and recorded in said history file.

17. The method of claim 16, wherein determination of which portion of data is to be restored, when complete restoration is not possible, is determined based on one or more criteria established in said history file.

18. A method of optimizing memory resources in a computing environment with at least one running application using message queuing, comprising:

establishing a history file containing a historical message processing rate for at least one application running in a computing environment;

storing said history file in a memory location in said computing environment;

selectively sweeping queuing information residing in or to be stored in said first memory to an alternate memory;

determining how much queuing information can be restored back to said first memory by unsweeping after comparing said history file including said historical message processing rate to a current message processing rate and determining current storage capacity in said first memory.

19. An apparatus for optimal information transfer in a computing environment, comprising:

a first memory operable to store processing information;

an alternate memory operable to store selectively reallocated processing information initially stored or scheduled to be stored on said first memory;

a history file containing historical processing information established about a computing environment or about at least one application running in said computing environment, said history file being stored selectively on said first memory or said alternate memory;

said computing environment operable to determine an amount of said reallocated processing to be restored back to said first memory based on historical processing information contained in said history file.

20. The method of Claim 19, wherein said computer environment comprises of at least one computer unit.